

VŠB – Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra informatiky

---

**Absolvovanie individuálnej odbornej  
praxe**  
**Individual Professional Practise in  
the Company**

2011

Pavol Kubjatko

Prehlasujem že som túto bakalársku prácu vypracoval samostatne. Uviedol som všetku použitú literatúru a publikácie.

V Ostrave dňa 27. marca 2011

.....

# **Abstrakt**

Cieľom tejto bakalárskej práce je oboznámiť vás, o absolvovaní odpornej individuálnej praxe vo firme XEVOS Solutions s.r.o.. V prvej kapitole sa zoznámime s firmou a firemným profilom. V druhej kapitole vás oboznámim so zadanými úlohami s, ktorými som sa stretol a, ako som ich riešil. V záverečnej kapitole zhodnotím svoje dosiahnuté skúsenosti a nové poznatky.

## **Kľúčové slová**

Visual Studio, MS SQL, LINQ, ASP .NET, CMS

# **Abstract**

The purpose of this Bachelors work is acquaint you about graduation individual professional practice in the Company Xevos Solutions Company Ltd.. In the first chapter you became acquainted with Companies profile. In the second chapter you will became acquainted with assigned tasks, with whom I have received and how I deal with them. In the final chapter I will evaluate my achieved experiences and new knowledges.

## **Key words**

Visual Studio, MS SQL, LINQ, ASP .NET, CMS

# **Zoznam použitých skratiek a symbolov**

ASP .NET – súčasť .NET Frameworku pre tvorbu webových aplikácií.

C#.NET – programovací jazyk

CMS – redakčný systém (Systém na správu webu, Administrácia)

DTO- data transfer objects je to návrhový vzor pre dizajn používaný na prenos dát medzi softwarovými aplikáciami alebo sa využíva na získavanie dát z databázy

Java Script – skriptovací jazyk

LINQ – dotazovací jazyk

Linqify - generátor DTO tried a dotazovací jazyk zložený na LINQ

MS SQL - databáza od Microsoftu

ORM – objektovo relačné mapovanie

Telerik – komponenty do Visual Studio

Visual Studio - vývojové prostredie od firmy Microsoft

# Obsah

1	Úvod .....	2
2	Informácie o Firme .....	3
2.1	Zoznámenie s firmou .....	3
2.2	Firemné zaradenie .....	3
3	Zadané úlohy a ich riešenie .....	4
3.1	Úloha generátor vlastného ORM.....	4
3.1.1	Riešenie .....	4
3.2	Projekt Laminas Hobby.....	5
3.2.1	Úvod .....	5
3.2.2	Úlohy .....	5
3.3	Cms Live.2.....	6
3.4	Projekt XevosWeb .....	7
3.4.1	Zadanie úlohy Novinky .....	7
3.4.2	Riešenie úlohy Novinky .....	7
3.5	Projekt Poptavka .....	8
3.5.1	Zadanie úlohy .....	8
3.5.2	Riešenie úlohy .....	8
4	Záver .....	12
5	Dosiahnuté skúsenosti .....	12
6	Použitá literatúra.....	13
7	Použitý kód .....	13
8	Zoznam obrázkov .....	13

# 1 Úvod

Pre vychádzajúceho študenta je veľmi ťažké nájsť si prácu. Väčšina firiem, ktoré sa nachádzajú v IT priemysle si žiada aspoň minimálnu prax v obore v ktorom chce uchádzač pracovať. Práve preto som si vybral namiesto bakalárskej práce individuálnu odbornú prax.

Ďalším dôvodom bolo zdokonalenie sa v danom obore. Naučiť sa nové technológie. Zistiť, aké je to pracovať v profesionálnom kolektíve programátorov. Taktiež zistiť, či mi bude v budúcnosti vyhovovať práca programátora a, či sa nedám na nejaký iný obor v IT priemysle.

## 2 Informácie o Firme

### 2.1 Zoznámenie s firmou

XEVOS Solutions poskytuje od svojho založenie v roku 2006 komplexné služby v oblasti informačných a komunikačných technológií s dôrazom na inovatívnosť a vysokú dostupnosť realizovaných riešení. Spoločnosť sa zameriava predovšetkým na konzultácie, realizácie a správu serverov, sieťovej infraštruktúry a klientskych staníc v segmente firiem pôsobiach na území celej Českej republiky.

V roku 2008 sa špecializácia firmy rozšírila o vývoj webových aplikácií na platforme ASP.NET, kde medzi pilotné projekty patrí vývoj webového ERP systému s pridanou hodnotou postaveného čisto ako webová aplikácia. Táto expanzia vyústila v roku 2008 založením spoločnosti XEVOS Solutions s.r.o. a posilnením tímu o nových členov. Dnes sa orientujú taktiež na Bussiness Intelligence a inovujú existujúce aplikácie o funkcie využívajúce BI.

### 2.2 Firemné zaradenie

Do firmy som bol prijatý na pozíciu vývojár ASP.NET v programovacom jazyku C#.NET. Túto pozíciu som vykonával takmer celú prax. V niektorých prípadoch som pracoval na desktopových projektoch v programovacom jazyku C#.NET.



Obrázok 1: Logo spoločnosti

## 3 Zadané úlohy a ich riešenie

### 3.1 Úloha generátor vlastného ORM

Ako prvú úlohu som dostal vytvoriť desktopovú aplikáciu, ktorá získa názvy všetkých tabuliek a názvy ich stĺpcov v databáze a vytvorí vlastné ORM. Použiť som mal programovací jazyk C# a vývojové prostredie Visual Studio 2010. Nadväzoval som na už vytvorený projekt „Stored Procedure Generator“, je to generátor uložených procedúr do SQL. Tento projekt vypracoval programátor, ktorý tu bol na praxi pred mnou.

#### 3.1.1 Riešenie

Začal som tým že som si preštudoval projekt na ktorý som mal nadväzovať. Potom som si urobil návrh a analýzu ako by sa dal rozšíriť tak aby som moc nezasahoval do už funkčného projektu, ale aby som dokázal čo najviac využiť už vytvorené metódy a triedy. Potom som začal s implementáciou. Najskôr som rozšíril formulárové okno o moje prvky, potrebné k realizácii daného projektu. Potom som si vytvoril vlastnú triedu, ktorú som naviazal na existujúce potrebné triedy. Potom som si z týchto tried získal pripojenie na databázu, pomocou príkazu SqlCommand a triedy ExecueReader som poslal na databázu požiadavku na vrátenie názvov všetkých tabuliek, atribútov a ich typu, ktoré som potom naukladal do listov<sup>1</sup>. Neskôr som tieto listy použil na vkladanie pomocou streamov, presnejšie pomocou triedy StreamWriter do súborov. Niektoré typy, ktoré som dostal z databáze som musel pretypovať napríklad som získal text a pretypoval som ho na string.

SQL Request<sup>2</sup>:

**Kod 1**

```
SELECT TABLE_NAME, COLUMN_NAME, DATA_TYPE,
CHARACTER_MAXIMUM_LENGTH FROM INFORMATION_SCHEMA.COLUMNS ORDR BY
TABLE_NAME
```

---

<sup>1</sup> Je to kolekcia určená pre zoskupovanie rôznych hodnôt, či už čísel alebo textov.

<sup>2</sup> Požiadavka na databázový server



## 3.2 Projekt Laminas Hobby

### 3.2.1 Úvod

Tento projekt mal obsahovať šesť webových stránok a vlastné CMS. Dostal som za úlohu implementovať triedy do CMS, ktoré budú slúžiť na správu webu. Cez tieto triedy sa dajú pridávať nové záznamy do tabuliek, ale taktiež sa dajú editovať, mazať poprípade sa dajú vypísať do gridu všetky záznamy z vybranej tabuľky. Samozrejme tieto záznamy sa dajú rôzne triediť napríklad podľa abecedy, podľa dátumu pridania a podľa mnohých iných atribútov. CMS bolo rozdelené na tri hlavné vrstvy BO, BL, DB. BO znamená „Business Objects“, v tejto vrstve boli triedy, ktoré som vytváral v projekte „Generátor vlastného ORM“. BL znamená „Business Logic“ v tejto vrstve sa nachádzajú triedy, ktoré prepájajú vrstvy BO a DB. Táto vrstva je nadriadená vrstvám BO a BD a iba s touto vrstvou programátor pracuje v iných triedach. DB znamená „Database“, v tejto vrstve sú triedy na prácu priamo z databázou. Nachádzajú sa tu dotazy na databázu.

K hotovému CMS má prístup iba admin a nie bežný užívateľ. Bežný užívateľ by o redakčnom systéme ani nemal vedieť.

### 3.2.2 Úlohy

Mojou prvou úlohou bolo implementovať triedu, v ktorej sa budú vkladať, upravovať, a mazať články. Dostal som prístup k už vytvorenej MS SQL databáze, kde som si našiel tabuľku Articles, nad ktorou som mal vykonávať SQL dotazy. Táto tabuľka bola úplne prázdna. Začal som implementáciou metódy na vkladanie záznamov. Najskôr som si vytvoril triedu v DB „addArticle“ v ktorej som si na implementoval SQL Dotaz na vkladanie do tabuľky za použitia ORM vrstvy z BO. Ďalej som vytvoril triedu v BL, aby som mohol pracovať z triedou v DB „addArticle“. Nazval som ju „addArticle“. Potom som si vo Visual Studiu vytvoril triedu „Articles.aspx“. V jej vizuálnej časti som s pomocou Telerik komponentmi vytvoril formu, ktorá obsahovala textboxy, tlačítka a popisy na vkladanie Článkov. Potom som vytvoril metódu, ktorá získavala hodnoty z týchto komponent a posielal ich na vrstvu BL do triedy „addArticle“, ktorá ich posielala do vrstvy BD do triedy „addArticle“. Táto trieda ich potom poslala na databázu a databáza tento záznam uložila. Potom som vytvoril takým istým spôsobom výpis všetkých článkoch a i potom aj ich editáciu. Všetky vkladania a výpisy z databázy boli riadené z jedinej triedy „CoreDB“. V tejto triede bolo pripojenie na databázu a metódy na vkladanie a výpis z databázy, teda napríklad „SqlDataReader“.

V prípade CMS sa nepredpokladá že by chcel niekto vkladať nejaký script, ktorý by škodil systému, preto sa do týchto formulárov nevkladá žiadna CAPTCHA<sup>3</sup> a teda ani žiadna ochrana proti scriptom.

Hneď ako som skončil s článkami som dokončil užívateľov „Users“, ktorých predtým rozrobil iný pracovník, ale bol presunutý na iný projekt. Bohužiaľ po dokončení užívateľoch som sa dozvedel, že je tento projekt pozastavený a nakoniec aj zrušený.

### 3.3 Cms Live.2

CMS Live.2 je nadväzujúci projekt na „Laminas Hobby“. Tento projekt bol založený, aby vytvoril kvalitné CMS, ktoré by bolo univerzálne pre viacero webov. V tomto projekte bol použitý na dotazovanie sa na databázu LINQ. Na vytvorenie Databázovej vrstvy bol použitá nadstavba „Linqify“. Linqify generuje DTO triedy. Takže sme ich nemuseli vytvárať ručne, čo nám uľahčilo prácu. Na druhú stranu sme sa ho museli naučiť používať, čo nám zabralo nejaký čas. V linqify môžete používať príkazy typu „Uzivatel.Save()“ na vloženie užívateľa do databázy, alebo `Uzivatel.Load(5)` pre načítanie piatich užívateľoch.

Mojou prvou úlohou bolo ako aj v projekte predtým vytvoriť triedy a metódy na manipuláciu s článkami. Metódy boli veľmi podobné ako v projekte „Laminas Hobby“. Rozdiel bol v tom, že na editovanie a vkladanie záznamov sme používali nové okno, ktoré bolo vytvorené javascriptom. Tam nastávali problémy so synchronizáciou napríklad keď som vytvoril nový záznam tak sa mal automaticky zobrazit' v gride na hlavnom okne. Tento problém bol neskôr vyriešený iným pracovníkom, ktorý na tom pracoval so mnou.

Nakoniec bol aj tento projekt pozastavený a zrušený. Následne bol vytvorený nový projekt Laminas Hobby 2 na tomto projekte som sa podieľal pri analýze a návrhu databázovej vrstvy. tento projekt bol rozbehnutý až na konci mojej praxe a tak som sa už nezúčastnil ďalšieho vývoja.

---

<sup>3</sup> Systém znakov ktorý nevie počítať, robot prečítať.

## 3.4 Projekt XevosWeb

Projekt Xevos Web je nový web pre firmu Xevos Solutions s.r.o.. Na tomto projekte som ja pracoval iba dodatočne. Opravoval som nejaké chyby, ktoré sa našli testérom, poprípade kóderom, ktoré on sám nevedel opraviť. Tento projekt je založený na UserControls, to znamená že obsah vytvárate v .ascx triedach, ktoré sa nedajú sami zobraziť ale musia sa vkládať do .aspx. tried.

### 3.4.1 Zadanie úlohy Novinky

Dostal som vytvoriť triedu pre databázovú tabuľku novinky.

### 3.4.2 Riešenie úlohy Novinky

Vytvoril som si triedu „News.ascx“, v tejto triede som si vytvoril repeater<sup>4</sup>, ktorý som naviazal na databázu pomocou triedy DataSource do, ktorej som vložil triedu getLastFourNews(). Táto trieda je z vrstvy bussiness logic a vracia posledné štyri novinky z databáze.

#### Kod 2 Repeater

```
Repeater1.DataSource = bl.getLastFourNews();
```

Ďalej som do repeateru vložil potrebné komponenty, ktoré som naviazal na prvky, ktoré som získal z databáze pomocou príkazu Eval(“názov atributu”).

---

<sup>4</sup> Umožňuje vytvárať ľubovoľný obsah z dát z databázy.

## 3.5 Projekt Poptavka

### 3.5.1 Zadanie úlohy

Vytvoriť HTML parser, ktorý bude sťahovať informácie z verejných štátnych výberových konaní a ukladať ich na databázu. Tento parser ma byť stále aktívny a v pravidelných intervaloch sa má spúšťať. Tabuľky sme dostali už vytvorené pretože už existoval systém ktorý tieto informácie zobrazoval. Zadane stránky [www.allytrade.cz](http://www.allytrade.cz), <https://gem.b2bcentrum.cz/softrade/>.

Tento systém bol vytvorený pretože spoločnosti, ktoré poskytujú tieto stránky odmietli dodať už hotový produkt. A odkázali firmu na párovanie HTML.

### 3.5.2 Riešenie úlohy

Začal som párovaním HTML [www.allytrade.cz](http://www.allytrade.cz). Zobrazil som si stránku z, ktorej som mal sťahovať informácie, kde som zistil, že táto informácie, ktoré budem potrebovať sú uložené v tabuľke a že tieto výberové konania sú na viacerých stránkach pomocou stránkovania takzvaný paging. Tabuľka obsahovala názvy výberových konaní, ktoré slúžili ako tlačidlá cez, ktoré sa dostanete na detail tohto výberového konania. V tomto detaile sa nachádzajú informácie, ktoré potrebujem.

Najskôr som potreboval zistiť linky na stránok zo stránkovania. Zistil som že tieto linky sú takmer také iste a líšia sa iba jedným číslom. Toto číslo je stránka na, ktorej sa nachádzame.

Vytvoril som Projekt Poptavka vo Visual Studiu 2010 v nej triedu Parser.cs. V tejto triede som importoval knižnice pre prácu s web stránkami. Na sťahovanie obsahu web stránok som použil WebClient. Vytvoril som si inštanciu na WebClient, ktorú som nazval client. Potom som vytvoril triedu GetPages ktorá mala návratovú hodnotu typu int. Táto metóda zisťovala počet stránok. V tejto metóde som vytvoril nekonečný while, ktorý obsahoval metódu client.DownloadString(string Url), táto metóda sťahuje obsah HTML. Url sú odkazy na ďalšie stránky zo stránkovania . Pomocou pomocnej premennej som zvyšoval číslo stránky v tejto url. Vždy keď sa zvýši číslo stránky sa stiahne celý obsah a zistí či ich obsahuje linky na detaily výberových konaní. Ak áno tak sa tieto linky pridajú do vytvoreného listu detailov. Ak neobsahuje žiadne, tak tato metóda skončí vráti počet stránok.

Na zisťovanie linkov detailov som vytvoril metódu Parse(string text). Ako vstupný parameter je text (HTML kód) stránky, ktorá obsahuje výberové konania. Výstupom tejto metódy je list detailov. Pomocou metód IndexOf() a Substring(), ktoré obsahuje string, sa hľadal zo stiahnutého textu časti textu, ktoré sa nachádzajú pred linkami detailov. Najskôr treba nájsť takú časť textu, ktorá je

jedinečná pre danú tabuľku. Potom časť textu ktorá sa opakuje pre všetky riadky tabuľky a potom časti textu medzi ktorými sa nachádza Url link detailu.

Príklad hľadania:

#### Kod 3 Hľadanie v stringu

```
int startIndex = part.IndexOf("href=");
int endIndex = part.IndexOf("'", startIndex + 7);
int length = endIndex - startIndex - 6;
string url = part.Substring(startIndex + 6, length);
```

Po zistení všetkých linkov detailov som si vytvoril metódu DownloadAndParse(string url), ktorá má vstupný parameter url čo je link pre detail výberového konania a návratovú hodnotu Poptavka. Táto metóda sťahuje pomocou WebClientu obsah detailov výberových konaní. Postupne prechádza celý text hľadá a sťahuje potrebné informácie. Napríklad názov spoločnosti, adresa spoločnosti, názov výberového konania a mnohé iné. Informácie ktoré sa majú sťahovať boli zadané pri zadávaní projektu. Po stiahnutí všetkých potrebných informácií sú tieto informácie zaslané na databázu.

Návratová hodnota Poptavka je trieda z ORM. Toto mapovanie som nevytváral ja ale môj kolega. ORM obsahuje metódy a triedy na prácu z databázou. Zasiela mnou získane atribúty na databázu, do tabuliek, ktoré sme dostali.

Neskôr som začal s parsovaním , <https://gem.b2bcentrum.cz/softrade/>. Tato stránka sa nedala parsovať pomocou triedy WebClient, pretože bola chránená. Musela byť použitá trieda HttpWebRequest, ktorá vie simulovať pripojenie internetovým prehliadačom. Bola vytvorená metóda GetPageContent0 tato metóda simuluje internetový prehliadač, sťahuje cookies v ktorých je uložené SessionID a sťahuje obsah stránky.

Podrobný popis metódy GetPageContent(string url)

Vytvorenie HttpWebRequestn a získanie obsahu:

#### Kod 4 HttpWebRequest

```
HttpWebRequest request = HttpWebRequest.Create(url) as HttpWebRequest;
```

Simulovanie internetového prehliadača:

#### Kod 5 HttpWebRequest

```
request.UserAgent = "Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US)
AppleWebKit/534.16 (KHTML, like Gecko) Chrome/10.0.648.205 Safari/534.16";
request.Accept = "text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8";
request.Method = "GET";
request.KeepAlive = true;
```

Vytvorenie kontajneru pre Cookies a získanie SessionID:

#### Kod 6 Cookies

```
CookieContainer cc = new CookieContainer();
cc.Add(new Cookie("JSESSIONID", m_sessionID, "/softtrade", "gem.b2bcentrum.cz"));
request.CookieContainer = cc;
```

Stiahnutie textu Html kódu:

#### Kod 7 Stiahnutie Html

```
using (WebResponse response = request.GetResponse())
using (Stream responseStream = response.GetResponseStream())
using (StreamReader reader = new StreamReader(responseStream))
{
    string responseFromServer = reader.ReadToEnd();
    return responseFromServer;
}
```

Po tom ako sa získa obsah stránky sa už postupuje takmer rovnako ako u predchádzajúceho parsovania s tým rozdielom že na sťahovanie jednotlivých linkov sa používa vždy metóda GetPageContent a sťahuje sa vždy iba prvá stránka výberových konaní, pretože sa zatiaľ nepodarilo prísť na to ako prejsť na ďalšiu stránku.

Po rozpársovaní týchto stránok som pridal k tomuto pridal Windows službu. Tato služba slúži na to aby tento program bol stále aktívny.

Na vytvorenie tejto služby som použil návod zo stránok MSDN odkaz nájdete v sekcii použitá literatúra. Najskôr som vytvoril projekt PoptavkaService. Pridal som k nemu inštalačný balík ku, ktorému som pridal konfiguračný súbor XML. Tento súbor obsahuje čas za ktorý sa ma služba spúšťať. Do projektu som pridal referencie z projektov na parsovanie a posielania dát na databázu.

Trieda PoptavkaService.cs sa vytvára automaticky. Táto trieda obsahuje metódy OnStart a OnStop. Do týchto metód sa dáva kód , ktorý sa ma vykonať keď sa služba spustí a keď skončí. Pridal som metódu ThreadStart, táto metóda obsahuje nekonečný cyklus while odkazy na spúšťanie parserov a metódu Thread.Sleep(čas), ktorá toto vlákno zastaví a znova spustí po určitej dobe. Čas sa

získava z metódy LoadXML, ktorá má návratovú hodnotu int. V tejto metóde sa načítava konfiguračné XML pomocou triedy XDocument. Cestu k tomuto xml dokumentu získame pomocou metódy AppDomain.CurrentDomain.BaseDirectory. Je to cesta, kde je nainštalovaná služba. Pomocou metódy GetElementsByTagName("Time") získame element Time, ktorý uložíme do XmlNodeList. Potom pomocou metódy Convert.ToInt32 a innerText získame čas, keďže je zadný čas v minútach musíme ho prerátať na milisekundy. V metóde OnStart sa metóda ThreadStart spúšťa pomocou vlákien.

#### Kod 8 XmlDocument

```
XmlDocument doc = new XmlDocument();  
doc.Load(AppDomain.CurrentDomain.BaseDirectory + "Config/Config.xml");  
XmlNodeList time = doc.GetElementsByTagName("Time");  
int cas = Convert.ToInt32(time[0].InnerText);  
cas = cas * 60000;  
return cas;
```

## 4 Záver

V priebehu praxe som pracoval na niekoľkých projektoch, ktoré pomohli firme pri vytváraní ďalších projektov. Projekt ORM sa používa v ďalších projektoch a uľahčuje prácu programátorom pri písaní zdĺhavých tried pre prácu z databázou. Projekt Poptávka sa začína využívať naplno a pridávajú sa tam stále nové stránky. Z projektov Laminas Hobby a CMS sa využívajú niektoré triedy, ktoré boli mnou vytvorené v projekte Laminas Hobby 2.

## 5 Dosiahnuté skúsenosti

Behom svojej praxe vo firme Xevos Solutions s.r.o. som sa naučil, aké to je pracovať v tíme. Naučil som sa nové technológie. Keďže pri nástupe na prax som mal minimálne vedomosti o javascripte, linqu a takmer žiadne o ASP.NET bolo pre mňa z počiatku ťažké pracovať na zadaných úlohach, ale v priebehu praxe som sa naučil používať tieto technológie a ich rôzne funkcie, ktoré uľahčujú prácu s týmito technológiami.



## 6 Použitá literatúra

**Linquify. 2011.** Linquify. *Linquify*. [Online] 23. Jul 2010. [Dátum: 4. Máj 2011.]  
<http://linquify.codeplex.com/>.

**MacDonald, Matthew, Freeman, Adam a Szpuszta, Mario. 2010.** *Pro ASP.NET 4 in C# 2010, Fourth Edition*. New York : Apress, 2010. 978-4302-2529-4.

**Microsoft. 2007.** MSDN. *Microsoft Support*. [Online] 26. November 2007. [Dátum: 4. Máj 2011.]  
<http://support.microsoft.com/kb/816169/en-us?fr=1>.

**Thau, Dave. 2009.** *Velký průvodce JavaScriptem*. Praha : PB tisk, 2009. 978-80-247-2211-5.

## 7 Použitý kód

Kod 1.....	4
Kod 2 Repeater .....	7
Kod 3 Hľadanie v stringu .....	9
Kod 4 HttpRequest.....	9
Kod 5 HttpRequest.....	10
Kod 6 Cookies .....	10
Kod 7 Stiahnutie Html .....	10
Kod 8 XmlDocument.....	11

## 8 Zoznam obrázkov

Obrázok 1: Logo spoločnosti .....	3
-----------------------------------	---